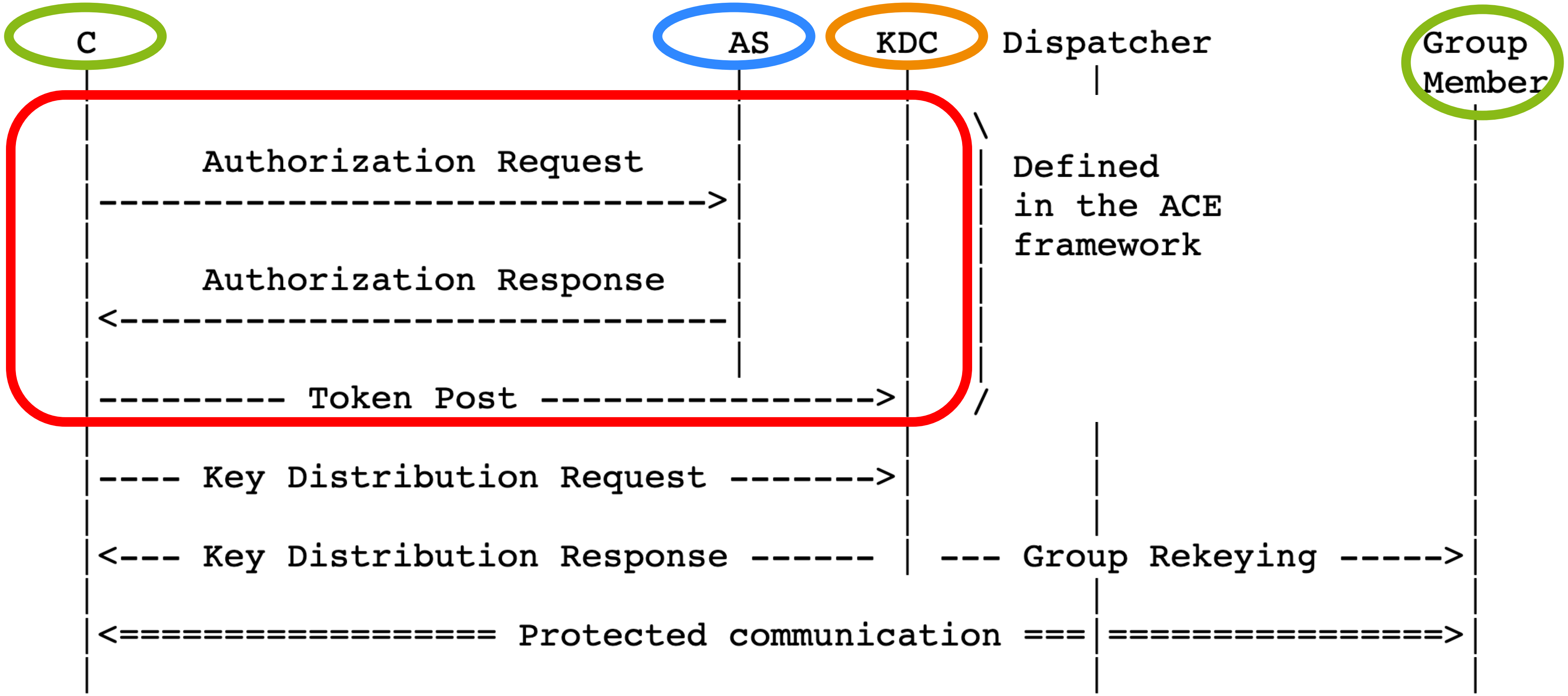
Key Provisioning for Group Communication using ACE

draft-ietf-ace-key-groupcomm-02

Francesca Palombini, Ericsson
Marco Tiloca, RISE

IETF 105, ACE WG, Montreal, July 25th, 2019

Quick Recap



```
C                             AS     KDC   Dispatcher          Group
|                              |       |        |              Member
|                              |       |  \                      |
|     Authorization Request    |       |  |  Defined             |
|----------------------------->|       |  |  in the ACE          |
|                              |       |  |  framework           |
|     Authorization Response   |       |  |                      |
|<-----------------------------|       |  |                      |
|                              |       |  |                      |
|--------- Token Post ----------------->| /                      |
|                                      |        |                |
|---- Key Distribution Request ------->|        |                |
|                                      |        |                |
|<--- Key Distribution Response ------ | --- Group Rekeying ----->|
|                                               |                 |
|<================== Protected communication ===|================>|
|                                               |                 |
```

Figure 2: Message Flow Upon New Node's Joining

Status Update v-02 – Feedback from WG?

Updated following review by Jim

- Added procedures for updating and renewing keying material
- Added a “*type*” to each “Key Distribution” request (Client to KDC):
  - *Key distribution* (= join) → when joining the group,
  - *Leave* → causes re-keying by the KDC for the current group members,
  - *Update* → node requests the current most updated keying material,
  - *New* → node requests that the KDC updates the node’s own individual keying material,
  - *Pub keys* → request one or more public keys of group members

- Add additional fields for the KDC telling the Client what type of public key to use (in the 2.01 response to token POST)
- Added PoP signature of a nonce from Client to KDC (in Key Distribution Request)
- Added requirements on profiles of this document (multicast, pubsub)
- Defines “application profile”
- Format of parameters
- IANA registration (labels, parameters that are CBOR maps)
- Expanded security considerations

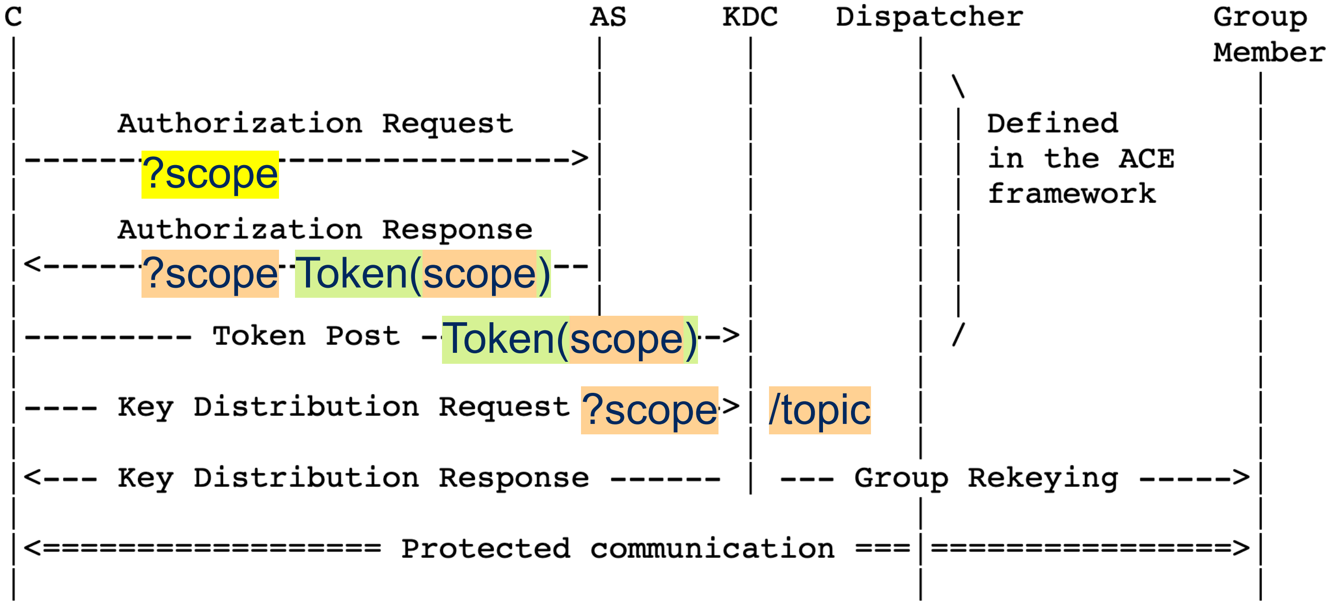
V-02 Reviews – Open point 1: scope

Scope: CBOR array of

- Group id (multicast) or topic (pubsub)
- Role(s)

Examples:

- CBOR: [topic1, [“publisher”, “subscriber”]]
- MQTT: text string “publish_topic1 subscribe_topic1/#”



```
C                            AS     KDC   Dispatcher        Group
|                             |      |        |             Member
|                             |      |        | \             |
|    Authorization Request    |      |        | | Defined     |
|------?scope---------------->|      |        | | in the ACE  |
|                             |      |        | | framework   |
|    Authorization Response   |      |        | |             |
|<-----?scope Token(scope)----|      |        | |             |
|                             |      |        | |             |
|--------- Token Post -Token(scope)->|        | /             |
|                                    |        |               |
|---- Key Distribution Request ?scope>| /topic |              |
|                                    |        |               |
|<--- Key Distribution Response ------| --- Group Rekeying ----->|
|                                             |               |
|<================= Protected communication ===|=============>|
|                                             |               |
```

Figure 2: Message Flow Upon New Node's Joining

“Should we allow multiple scopes in the same access token?”

- If group id, no, because there is a 1-to-1 mapping between scope and security group
- If topic, yes, you could want same security material for different topics

V-02 Reviews – Open point 1: scope

Scope: CBOR array of

- Group id (multicast) or topic (pubsub)
- Role(s)

Format of ‘scope’:

- CBOR: [[topic1], [“publisher”, “subscriber”]]
- MQTT: text string “publish_topic1 subscribe_topic1/#”

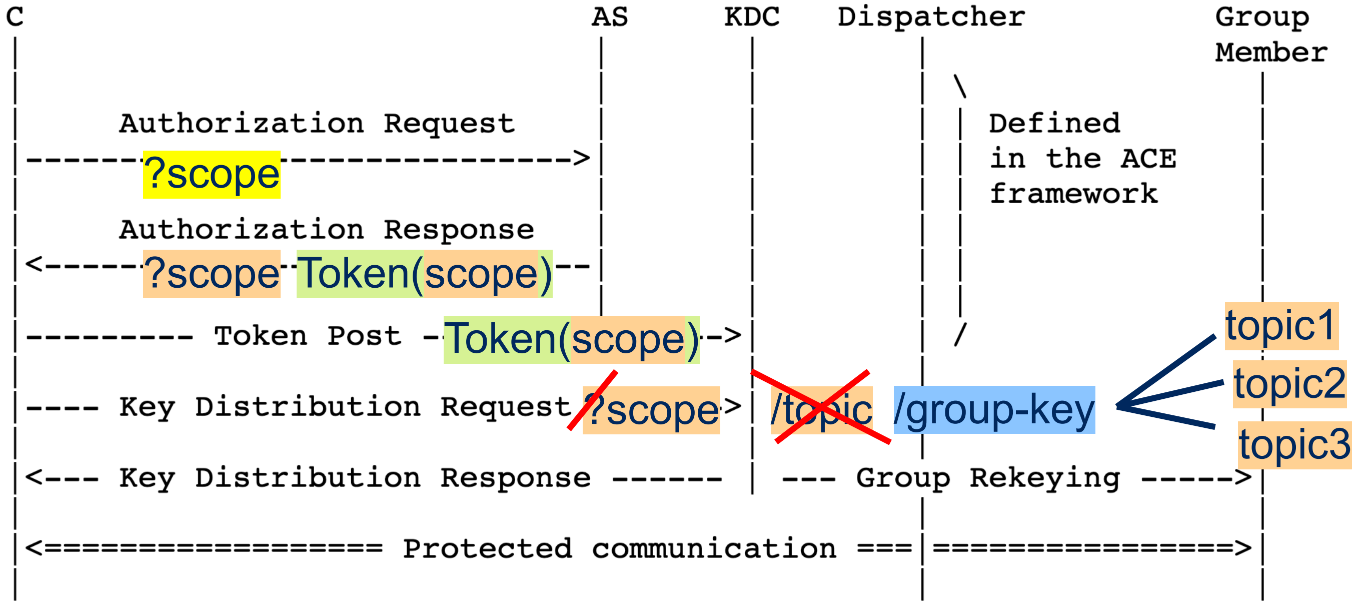


Figure 2: Message Flow Upon New Node's Joining

Proposal:

- Allow multiple scopes (i.e. group ids or topics), which can be useful in pubsub
- Key Distribution Request is sent to a fixed “group Uri path” rather than one associated with scope
- ‘scope’ becomes mandatory in Key Distribution Request

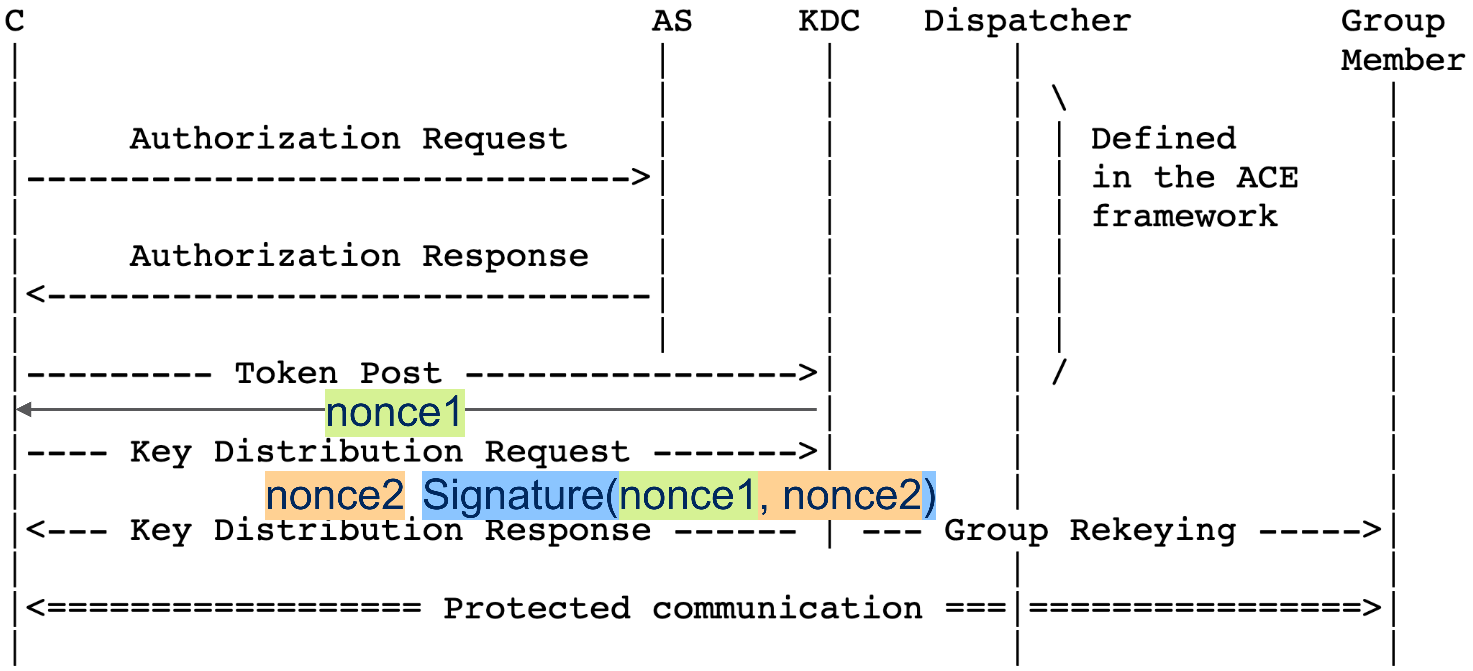
V-02 Reviews – Open points

- Right now C signs a nonce (nonce1) generated by KDC, for PoP of pub key

Proposal: Add: C generates a second nonce (nonce2), signs it together with nonce1, sends it to KDC in Key Distribution Request

- Wrong registration of new parameters (AS creation hints)

Proposal: fix that, register in “Oauth Parameters” and “OAuth Parameters CBOR Mappings”?



```
C                              AS      KDC    Dispatcher          Group
|                              |        |         |               Member
|                              |        |         | \                |
|     Authorization Request    |        |         | | Defined        |
|----------------------------->|        |         | | in the ACE     |
|                              |        |         | | framework      |
|     Authorization Response   |        |         | |                |
|<-----------------------------|        |         | |                |
|                              |        |         | |                |
|--------- Token Post ------------------>|         | /                |
|<--------------nonce1-------------------|         |                  |
|---- Key Distribution Request -------->|         |                  |
|          nonce2 Signature(nonce1, nonce2)        |                  |
|<--- Key Distribution Response ------- | --- Group Rekeying ----->|
|                                                  |                  |
|<================== Protected communication ===|=================>|
|                                                  |                  |
```

Figure 2: Message Flow Upon New Node's Joining

- Format of keys? (for encryption key, and public keys used)

Proposal: define all keys as using the format of the keys in ‘cnf’

Other – Open points

- Rekeying: “*KDC should renew the keying material upon group membership change, and should provide it to the current group members through the rekeying scheme used in the group.”*
- Right now: we define how the C can get the new keying material. “*Alternatively, the re-distribution of keying material can be initiated by the KDC”*
- What endpoint does KDC use to send rekeying messages? Is this in scope of this doc?

Proposal: Define new (optional) parameter ‘rekeying_uri’ in the Key Distribution Request. C use this parameter to tell the KDC what uri to use for unicast rekeying messages.

- One of the alternatives mentioned for rekeying is with multicast messages.
- Client would not know what IP multicast address to listen to for rekeying.

Proposal: Define new (optional) parameter ‘rekeying_uri’ in the Key Distribution Response. C use this parameter to listen for rekeying messages.